Advanced Deep Learning
AIGC 5500
Midterm Project
Deep Learning Optimizers

1. Introduction

- **Objective**: Research and investigate to compare the performance of **Adam** (Adaptive Moment Estimation), **RMSprop** (Root Mean Square Prop), and **AdamW** (Adam with Weight Decay) optimizers on a feedforward fully connected neural network using the KMNIST dataset.
- **Importance**: Understanding the strengths and weaknesses of different optimization algorithms helps in selecting the right one for specific tasks in deep learning.

2. Dataset Description

- **KMNIST Dataset**: A dataset of handwritten Japanese characters, like MNIST but more complex.
    - **Training Set**: 60,000 images
    - **Test Set**: 10,000 images
    - **Image Size**: 28x28 pixels, grayscale

3. Deep Learning Model

- **Architecture**: Design a feedforward fully connected neural network.
    - **Input Layer**: 784 neurons (28x28 pixels)
    - **Hidden Layers**: Two hidden layers with 128 and 64 neurons respectively
    - **Output Layer**: 10 neurons (one for each class)
    - **Activation Function**: Use ReLU for hidden layers and SoftMax for the output layer.
    - Use Cross-Entropy Loss Function

4. Methodology

- **Hyperparameter Tuning**: Use a systematic search to find the best hyperparameters for each optimizer.
- **Cross-Validation**: Implement 5-fold cross-validation to ensure robust evaluation.
- **Training and Evaluation**:
    - Train the model using each optimizer.
    - Evaluate performance on training, validation, and test datasets.
    - Record metrics such as accuracy, loss, and training time.

5. Results

- **Tabular and Graphical Representation**:
    - Create tables showing accuracy, loss, and training time for each optimizer.
    - Generate graphs comparing the performance metrics across different optimizers.

6. Interpretation and Discussion

- **Analysis**: Discuss the performance of each optimizer, highlighting strengths and weaknesses.
- **Conclusion**: Summarize findings and suggest the best optimizer for this specific task.

7. References

- Cite all resources and papers used in the project.

Additional Instructions

- **Code Documentation**: Ensure your code is well-documented with comments explaining each part.

- **Readme File**: Provide clear instructions on how to set up the environment and run the code.
- **Version Control**: Use version control (e.g., Git) to manage your project files and collaborate with team members.

Deliverables:

- **PDF Report**:
  - Project Introduction
  - Dataset, Model, and **Optimizers Description**
  - Solutions, Findings, and Results
  - Interpretation, Discussion, and Conclusion
  - Each Member's Contributions
  - References
- **A Video Presentation Link (15 to 20 Minutes):**
  - Create a PowerPoint-based video presentation explaining your project, including your solutions, findings, and results.
    - ⚠ Do not include code explanations in the video.
  - Any number of students from your group may participate in the presentation.
  - Upload your video to Humber OneDrive and share the link with me.
    - 🔑 You are responsible for ensuring the link is accessible.
- **A Zip File** (**no RAR, no 7z, etc.**) containing ALL **Python Files:**
  - .py and .ipynb files with code and results
  - Readme file with instructions on how to run the code